Симметричные семейства алгоритмов

Помимо вычисления точных комбинаторных оценок вероятности переобучения для конкретных семейств алгоритмов большой интерес представляют универсальных верхние оценки, справедливые для произвольных семейств алгоритмов. В статье приводится пример подобной оценки, которая оказывается точной для монотонных цепочек и монотонных сеток произвольной размерности. Поиск способов устранения завышенности данной оценки приводит к определению группы симметрий семейства алгоритмов и определению идентичных алгоритмов. Доказывается, что идентичные алгоритмы имеют равную вероятность реализоваться в результате обучения и дают равный вклад в вероятность переобучения. На основании этой теоремы вычисляется точная оценка вероятности переобучения для единичной окрестности лучшего алгоритма, монотонной и унимодальной цепочек, слоя алгоритмов. В заключении приводятся примеры вычисления групп симметрии конкретных семейств алгоритмов. Также затрагивается вопрос об использовании неравенства Коши-Буняковского-Шварца в комбинаторной теории оценки обобщающей способности.

Содержание

1 Универсальная верхняя оценка вероятности переобучения **2**

2 Семейства алгоритмов **3**

3 Вероятность переобучения **3**

4 Группа симметрии семейства алгоритмов **4**

5 Точные оценки вероятности переобучения для модельных семейств алгоритмов **6**

5.1 Единичная окрестность лучшего алгоритма 6
5.2 Унимодальное семейство A_D 7
5.3 Монотонная цепочка 8
5.4 Семейство из p монотонных цепочек, склеенных в лучшем алгоритме 8
5.5 Семейство «Полный слой алгоритмов» 9

6 КБШ-оценка вероятности переобучения **10**

7 Граф смежности семейства алгоритмов **11**

8 Вычисление групп симметрии **12**

1 Универсальная верхняя оценка вероятности переобучения

Помимо вычисления точных комбинаторных оценок вероятности переобучения для конкретных семейств алгоритмов большой интерес представляют универсальных верхние оценки, справедливые для произвольных семейств алгоритмов. Для получения такой оценки воспользуемся тем же подходом, который позволяет вычислять точные оценки для конкретных семейств. А именно, укажем способ построения множества производящих и разрушающих объектов для произвольного семейства алгоритмов.

Напомним теорему о точной оценке вероятности переобучения.

Теорема. Пусть множество A, выборка $\mathbb{X}$, и метод μ таковы, что для каждого алгоритма можно указать пару подмножеств $X_a \subset \mathbb{X}$ и $X_a' \subset \mathbb{X}$, удовлетворяющую условию

$$\{\mu X = a\} \leftrightarrow \{X_a \subset X\} \text{ и } \{X_a' \subset \bar{X}\},\ \forall X \in [\mathbb{X}]_L^\ell$$

Тогда можно выписать точное выражение для вероятности переобучения в явном виде

$$Q_\varepsilon = \sum_{a \in A} P_a H_{L_a}^{\ell_a, m_a}(s_a(\varepsilon)).$$

Данная теорема требует, что бы условие было необходимым и достаточным. Однако уже для унимодальных цепочек и сеток предъявить подобный критерий получения алгоритма не просто. Заметим, что ослабив требования теоремы, мы получим простую и универсальную верхнюю оценку для вероятности переобучения.

Следствие. Пусть $\{X_a \subset X\}$ и $\{X_a' \subset \bar{X}\}$ будут необходимыми, но уже не обязательно достаточным условием получения алгоритма в результате обучения. Тогда явная формула предыдущей теоремы будет давать верхнюю оценку вероятности переобучения.

Осталось лишь указать способ построения множеств X_a и X_a' для произвольного семейства алгоритмов.

Множество $\bar{A}_a \subset \mathbb{A}$ — это множество алгоритмов, чье множество ошибок содержится среди множества ошибок алгоритма a. Это — строго более хорошие алгоритмы. Определим множество всех объектов, на которых алгоритм ошибается, а хотя бы один $a' \in \bar{A}_a$ — нет:

$$X_a' = \bigcup_{a' \in \bar{A}_a} a \backslash a'$$

Для $\mu X = a$ необходимо, что бы выполнялось $X_a' \subset \bar{X}$.

Аналогично построим множество строго более плохих алгоритмов, делающих ровно на **одну** ошибку больше: $A_a \in \mathbb{A}$ и множество всех объектов, на которых a не ошибается, а хотя бы один из $a' \in A_a$ — ошибается:

$$X_a = \bigcup_{a' \in A_a} a' \backslash a$$

Для $\mu X = a$ необходимо что бы выполнялось $X_a \subset X$.

Заметим, что между построенными множествами нет естественной симметрии — множество X_a зависит только от строго на единицу худших алгоритмов, в то время как X_a' зависит только от всех строго лучших алгоритмов.

Таким образом, у нас есть универсальный способ построения множеств разрушающий и производящих объектов для произвольных семейств, и следствие теоремы, которое утверждает, что полученная оценка будет верхней. Легко доказать, что данная оценка является точной для монотонной цепочки. В работе [...] доказывается, что приведенная

выше оценка будет точна и для многомерной монотонной сетки алгоритмов произвольной размерности.

К сожалению, уже для единичной окрестности лучшего алгоритма приведенная оценка является сильно завышенной. При этом очевидно, что в этом семействе все алгоритмы, кроме лучшего, в определенном смысле одинаковы, неотличимы друг от друга. Это наблюдение наводит на мысль, что изучение свойств симметрии самого семейства алгоритмов позволит улучшить приведенную выше оценку, или получить принципиально новые способы оценивания вероятности переобучения.

2 Семейства алгоритмов

Алгоритмом $a \in 2^L$ мы будем называть бинарный вектор длины L. Рассмотрим произвольный *набор* алгоритмов $A = \{a_1, a_2, \ldots, a_D\}$. Его можно представлять себе как матрицу, строки которой соответствуют алгоритмам, а столбцы — объектам. При этом нужно иметь в виду, что при перестановке строк таблицы (т.е. алгоритмов) мы продолжаем иметь дело с тем же самым набором алгоритмов.

Обозначим множество всех алгоритмов за $\mathbb{A}$, множество всех наборов алгоритмов — $\aleph$. Его элементы будем обозначать заглавными латинскими буквами: $A \in \aleph, A \subset \mathbb{A}$. Заметим, что $|\aleph| = 2^{2^L}$. Действительно, количество подмножеств произвольного конечного множества X составляет $2^{|X|}$ элементов. Взяв в качестве X множество $\mathbb{A}$ получаем требуемый результат.

Обычно в задачах классификации порядок следования объектов не имеет значения. Поэтому наборы алгоритмов, которые можно получить друг из друга перестановкой объектов, будем называть *неразличимыми*. Очевидно, что неразличимость является отношением эквивалентности.

Определение. Семейство алгоритмов — это класс эквивалентности неразличимых наборов алгоритмов.

Будем обозначать элементы группы перестановок $\pi \in S_L$. Эта группа действует на множестве всех наборов алгоритмов перестановками столбцов матрицы. Семейства алгоритмов — это орбиты множества $\aleph$, при определенном выше действии группы перестановок S_L.

Задача. Выписать все семейства для L=3 (на единичном кубе).

Задача. Сколько существует различных семейств для данного L?

3 Вероятность переобучения

Метод обучения позволяет выбрать алгоритм из набора при разбиении выборки на обучение и на контроль. Пусть метод обучения использует только данные о числе ошибок на обучении. Тогда в тех ситуациях, когда нашлось несколько алгоритмов с равным числом ошибок на обучении, необходимо применять рандомизацию. Таким образом, метод обучения ставит в соответствие каждому алгоритму вероятность получить этот алгоритм в результате обучения.

Обозначим $[X]^\ell_L$ — множество всех ℓ-элементных подмножеств $\mathbb{X} = \{1, 2, \ldots, L\}$. Через $n(a, X) = \sum_{x \in X} a(x)$ обозначим число ошибок алгоритма на множестве $X \in [X]^\ell_L$. Частотой ошибок, или эмпирическим риском алгоритма a на выборке $X \in [X]^\ell_L$ называется величина $\nu(a, X) = \frac{1}{|X|} n(a, X)$. Дополнение будем обозначать с помощью черты над множеством: $\bar{X} = \mathbb{X} \backslash X$. Введем для краткости обозначение для разности частот ошибок алгоритма на контрольной и обучающей выборке: $\Delta\nu(a, X) = \nu(a, \bar{X}) - \nu(a, X)$.

Методом обучения будем называть произвольное отображение

$$\mu : \aleph \times [X]_L^\ell \times \mathbb{A} \to [0,1]$$

удовлетворяющее при всех $A \in \aleph$, $X \in [X]_L^\ell$ следующим условиям:

$$\sum_{a \in A} \mu(A, X, a) = 1,$$
$$\forall a_1, a_2 \in A \,|\, n(a_1, X) = n(a_2, X) \to \mu(A, X, a_1) = \mu(A, X, a_2),$$
$$\forall \pi \in S_L \to \mu(A, X, a) = \mu\left(\pi(A), \pi(X), \pi(a)\right).$$

Первое условие означает нормировку вероятности, и автоматически обеспечивает выполнение условия $\mu(A, X, a) = 0$, $\forall a \notin A$. Второе условие говорит, что при любом разбиении данных на наблюдаемую и скрытую выборку вероятность получить алгоритм в результате обучения зависит только от количества ошибок алгоритма на обучении. Третье условие означает, что метод обучения не учитывает порядок следования объектов в выборке.

Вероятностью получить алгоритм в результате обучения назовем величину

$$P_\mu(A, a) = \frac{1}{C_L^\ell} \sum_{X \in [X]_L^\ell} \mu(A, X, a)$$

Зафиксируем $\varepsilon \in [0, 1)$. Тогда для каждого набора алгоритмов можно вычислить вероятность переобучения:

$$Q_\mu(A) = \frac{1}{C_L^\ell} \sum_{X \in [X]_L^\ell} \sum_{a \in A} \mu(A, X, a) \left[\Delta\nu(a, X) \geq \varepsilon\right]$$

Покажем, что данное определение имеет смысл и для семейства алгоритмов. Для этого заметим, что применение произвольной перестановки к множеству разбиений на обучение и на контроль просто переставляет элементы этого множества: $\pi([X]_L^\ell) = [X]_L^\ell$, а так же воспользуемся свойством частоты ошибок алгоритма $\nu(\pi(a), \pi(X)) = \nu(a, X)$.

$$\begin{aligned}
Q_\mu(\pi(A)) =& \frac{1}{C_L^\ell} \sum_{X \in [X]_L^\ell} \sum_{a \in \pi(A)} \mu(\pi(A), X, a) \left[\Delta\nu(a, X) \geq \varepsilon\right] = \\
=& \frac{1}{C_L^\ell} \sum_{X \in \pi([X]_L^\ell)} \sum_{a \in \pi(A)} \mu(\pi(A), X, a) \left[\Delta\nu(a, X) \geq \varepsilon\right] = \\
=& \frac{1}{C_L^\ell} \sum_{X \in [X]_L^\ell} \sum_{a \in A} \mu(\pi(A), \pi(X), \pi(a)) \left[\Delta\nu(\pi(a), \pi(X)) \geq \varepsilon\right] = \\
=& \frac{1}{C_L^\ell} \sum_{X \in [X]_L^\ell} \sum_{a \in A} \mu(A, X, a) \left[\Delta\nu(a, X) \geq \varepsilon\right] = Q_\mu(A).
\end{aligned}$$

Совершенно аналогично можно показать, что вероятность $P_\mu(A, a)$ зависит от семейства, а не от набора алгоритмов.

4 Группа симметрии семейства алгоритмов

Определение. *Группой симметрий* набора алгоритмов A будем называть подгруппу группы перестановок $G \in S_L$, сохраняющую данный набор алгоритмов.

$$S(A) = \{\pi \in S_L : \pi(A) = A\} = Stab_A$$

Заметим, что определение группы симметрий дано для набора алгоритмов, а не для семейства. Покажем, что группы симметрий различных представителей одного семейства изоморфны.

Теорема. Пусть набор алгоритмов $A_2 = \pi(A_1)$, и пусть G_i — группа симметрий набора A_i, при $i = 1, 2$. Тогда группы G_1 и G_2 получаются друг из друга сопряжением: $G_2 = \pi \circ G_1 \circ \pi^{-1}$.

Таким образом, можно говорить о группе симметрий семейства алгоритмов.

Теорема. Группа симметрии множества алгоритмов изоморфно вкладывается в группу автоморфизмов соответствующего графа смежности.

С помощью группы симметрий семейства алгоритмов легко ответить на вопрос: сколько представителей может быть у данного семейства?

Теорема. Число наборов алгоритмов, соответствующих семейству A, равно $\frac{L!}{\#S(A)}$

Легко доказать, что группа симметрий монотонной цепочки тривиальна. Для унимодальной цепочки с равными длинами левой и правой ветви получим группу $S_2 = \mathbb{Z}/2\mathbb{Z}$, для единичной окрестности лучшего алгоритма — группу перестановок S_D, где D — число алгоритмов, за исключением лучшего. Легко построить семмейство алгоритмов с циклической группой симметрий $\mathbb{Z}/n\mathbb{Z}$ произвольного порядка:

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Задача. Вычислить группы симметрий шара алгоритмов, сетки и монотонной сетки алгоритмов, дерева с фиксированным коэффициентом ветвления.

Определение. *Идентичные алгоритмы* в семействе — это алгоритмы, лежащие в одной орбите действия группы симметрий. Т.е. $a_1 \sim a_2$ если $\exists \pi \in S(A) : \pi(a_1) = a_2$.

Теорема. Любая пара идентичных алгоритмов имеет равное число ошибок на полной выборке.

Неверно утверждать, что для идентичных алгоритмов a_1 и a_2 при всех $X \in [X]^\ell_L$ выполнено $\mu(A, X, a_1) = \mu(A, X, a_2)$. Тем не менее справедлива следующая теорема:

Теорема. Вероятность получить идентичные алгоритмы в результате обучения одинакова.

$$\begin{aligned} P_\mu(\pi(a), A) &= P_\mu(\pi(a), \pi(A)) = \\ &= \frac{1}{C^\ell_L} \sum_{X \in [X]^\ell_L} \mu(\pi(A), X, \pi(a)) = \\ &= \frac{1}{C^\ell_L} \sum_{X \in \pi([X]^\ell_L)} \mu(\pi(A), X, \pi(a)) = \\ &= \frac{1}{C^\ell_L} \sum_{X \in [X]^\ell_L} \mu(\pi(A), \pi(X), \pi(a)) = \\ &= \frac{1}{C^\ell_L} \sum_{X \in [X]^\ell_L} \mu(A, X, a) = P_\mu(a, A) \end{aligned}$$

Обозначим через $\Omega(A)$ множество классов идентичных алгоритмов, за $a_\omega \in A$ — произвольного представителя класса $\omega \in \Omega(A)$.

Теорема. Идентичные алгоритмы дают равный вклад в вероятность переобучения.

$$Q_\mu(A) = \frac{1}{C^\ell_L} \sum_{X \in [X]^\ell_L} \sum_{\omega \in \Omega(A)} |\omega| \, \mu(A, X, a_\omega) \left[\Delta\nu(a_\omega, X) \geq \varepsilon\right].$$

Доказательство. Распишем вероятность переобучения:

$$\begin{aligned}Q_\mu(A) =& \frac{1}{C_L^\ell} \sum_{X\in[X]_L^\ell} \sum_{a\in A} \mu(A,X,a)\,[\Delta\nu(a,X)\geq\varepsilon] = \\ =& \frac{1}{C_L^\ell} \sum_{X\in[X]_L^\ell} \sum_{\omega\in\Omega(A)} \sum_{a\in\omega} \mu(A,X,a)\,[\Delta\nu(a,X)\geq\varepsilon] = \\ =& \frac{1}{C_L^\ell} \sum_{\omega\in\Omega(A)} \sum_{a\in\omega} \underbrace{\sum_{X\in[X]_L^\ell} \mu(A,X,a)\,[\Delta\nu(a,X)\geq\varepsilon]}_{N(a)}\end{aligned}$$

В выделенном в данной формуле выражении $N(a)$ алгоритм a можно заменить на любой идентичный: $N(a) = N(\pi(a))$, где $\pi \in S(A)$. Таким образом вместо суммирования по $a \in \omega$ можно написать $|\,\omega|$ — количество идентичных алгоритмов. ■

5 Точные оценки вероятности переобучения для модельных семейств алгоритмов

5.1 Единичная окрестность лучшего алгоритма

Методом минимизации эмпирического риска мы будем называть следующий метод обучения. Выделим множество объектов, неразличимых на обучающей выборке, и допускающих при этом минимально-возможное число ошибок:

$$E_{A,X} = \operatorname*{Argmin}_{a\in A} n(a,X)$$

В результате обучения методом минимизации эмпирического риска все алгоритмы вне этого множества имеют нулевую вероятность реализоваться:

$$\mu_0(A,X,a) = \begin{cases} \frac{1}{|E_{A,X}|}, \forall a \in E_{A,X} \\ 0, \forall a \notin E_{A,X} \end{cases}$$

Найдем оценку вероятности переобучения для единичной окрестности лучшего алгоритма. Обозначим через D — число алгоритмов с одной ошибкой, m — число ошибок лучшего алгоритма, L — длину выборки, ℓ — кол-во объектов на обучении, k — на контроле. Введем обозначение a_0 для лучшего алгоритма семейства и $a_1, \ldots, a_D$ для его окрестности. Обозначим соответственно через $x_1, \ldots, x_D$ объекты, на которых различаются a_i и a_0. Объекты, на которых ошибаются все алгоритмы, будем называть шумовыми. Через z обозначим число объектов, на которых не ошибается ни один алгоритм.

Воспользуемся теоремой о равном вкладе идентичных алгоритмов в вероятность переобучения. Заметим, что в нашем семействе два класса идентичных алгоритмов: $\omega_1 = \{a_0\}$ и $\omega_2 = \{a_1, \ldots, a_D\}$. Поэтому в сумме $\sum_{\omega\in\Omega(A)}$ только два слагаемых. Выпишем их явно:

$$\begin{aligned}Q_\mu(A) = \frac{1}{C_L^\ell} \sum_{X\in[X]_L^\ell} \sum_{\omega\in\Omega(A)} |\,\omega|\,\mu(A,X,a_\omega)\,[\Delta\nu(a_\omega,X)\geq\varepsilon] = \\ \frac{1}{C_L^\ell} \sum_{X\in[X]_L^\ell} \mu(A,X,a_0)\,[\Delta\nu(a_0,X)\geq\varepsilon] + \\ \frac{D}{C_L^\ell} \sum_{X\in[X]_L^\ell} \mu(A,X,a_1)\,[\Delta\nu(a_1,X)\geq\varepsilon]\,.\end{aligned}$$

Введем параметр t — число объектов из множества $x_1, \ldots, x_D$, попавших в обучение. s — число шумовых объектов, попавших в обучение.

Вклад лучшего алгоритма в переобучение:

$$Q_\mu(A, a_0) = \frac{1}{C_L^\ell} \sum_{t=0}^{D} \sum_{s=0}^{s_0} \frac{1}{1+D-t} C_D^t C_m^s C_z^{\ell-t-s}$$

Вклад алгоритма a_1 в переобучение

$$Q_\mu(A, a_1) = \frac{1}{C_L^\ell} \sum_{t=0}^{D-1} \sum_{s=0}^{s_1} \frac{1}{1+D-t} C_{D-1}^t C_m^s C_z^{\ell-t-s}$$

Тут $s_0 = \frac{\ell}{L}(m - \varepsilon k)$, $s_1 = \frac{\ell}{L}(m + 1 - \varepsilon k)$.

Суммируя эти две оценки получаем искомую вероятность переобучения $Q_\mu(A, a_0) + DQ_\mu(A, a_1)$. ■

5.2 Унимодальное семейство A_D

Обозначим за m число ошибок лучшего алгоритма, D — длины ветвей (таким образом, что бы у последнего алгоритма обеих ветви число ошибок составило $m+D$). Группа симметрий данного семейства позволяет отождествлять пары алгоритмов с равным числом ошибок, таким образом вероятность переобучения можно записать в виде

$$Q_\varepsilon(A_D) = \sum_{h=0}^{D} |\omega_h| \sum_{t=h}^{D} \sum_{t'=0}^{D} \sum_{X \in N(t,t')} \mu(A, X, a_h) \left[\Delta\nu(a_h, X) \geq \varepsilon\right].$$

Индекс h обозначает номер класса эквивалентных алгоритмов (таким образом, что бы все алгоритмы класса ω_h имели $m + h$ ошибок). Следовательно $|\omega_h| = 1$ при $h = 0$ и $|\omega_h| = 2$ при $h \geq 1$. Зафиксировав индекс h мы можем считать, что исследуем вероятность переобучения алгоритма a_h, находящегося — для определенности — в левой ветви.

Индексы t и t' параметризуют состав множества неразличимых алгоритмов $E_{A,X}$. Для произвольного разбиения $X \in [\mathbb{X}]_L^\ell$ определим число t как максимальное число, для которого все объекты $x_1, x_2, \ldots, x_t$ находятся в контроле, а x_{t+1} (при его наличии) — в обучении. Индекс t' определяется аналогично, но с помощью объектов правой ветви. Фигурирующее в формуле множество $N(t, t')$ обозначает множество разбиений выборки с фиксированными параметрами t и t'.

Из определения t и t' следует, что $|E_{A,X}| = \frac{1}{1+t+t'}$. Единица в знаменателе относится к алгоритму a_0. Индексы t и t' при суммировании пробегают разные множества значений, поскольку при $t < h$ алгоритм a_h не лежит в множестве $E_{A,X}$, и метод обучения дает ему нулевой вес: $\mu(A, X, a_h) = 0$.

Осталось вычислить число выборок в множестве $N(t, t')$, на которых алгоритм a_h оказывается переобученным. Пусть s_0 — максимальное число ошибок на обучении, при котором переобучение все еще наблюдается. Легко вывести, что $s_0 = \lfloor \frac{\ell}{L}(m + h - \varepsilon k) \rfloor$.

Обозначим $L' = L - t' - t - 2$, $\ell' = \ell - 2$.

Нам необходимо из L' объектов выбрать ℓ' для обучения таким образом, что бы из m свободных ошибок алгоритма a_h в обучении оказалось не более s_0 ошибок. Это можно сделать $\sum_{s=0}^{s_0} C_m^s C_{L'-m}^{\ell'-s}$ числом способов.

Собирая все результаты, приходим к окончательной формуле:

$$Q_\varepsilon(A_D) = \sum_{h=0}^{D} |\omega_h| \sum_{t=h}^{D} \sum_{t'=0}^{D} \frac{1}{1+t+t'} \frac{\sum_{s=0}^{s_0} C_m^s C_{L'-m}^{\ell'-s}}{C_L^\ell}.$$

В данной формуле мы пренебрегли эффектами, связанными с нижними краями цепочки. В действительности следовало бы учесть, что $\ell' = \ell - [t \neq D] - [t' \neq D]$, и аналогично для L'. ■

Заметим, что данное рассуждение легко распространяется на произвольное количество «хвостов» у цепочек.

5.3 Монотонная цепочка

Упрощая рассуждения, приведенные для унимодальной цепочки, получим формулу

$$Q_\varepsilon(A_D) = \sum_{h=0}^{D} \sum_{t=h}^{D} \frac{1}{1+t} \frac{\sum_{s=0}^{s_0} C_m^s C_{L'-m}^{\ell'-s}}{C_L^\ell},$$

где $L' = L - t - [t \neq D]$, $\ell' = \ell - [t \neq D]$, $s_0 = \lfloor \frac{\ell}{L}(m + h - \varepsilon k) \rfloor$.

5.4 Семейство из p монотонных цепочек, склеенных в лучшем алгоритме

Естественным образом обобщая рассуждения, приведенные для унимодальной цепочки, получаем формулу

$$Q_\varepsilon(A) = \sum_{h=0}^{D} |\omega_h| \sum_{t_1=h}^{D} \sum_{t_2=0}^{D} \dots \sum_{t_p=0}^{D} \frac{1}{1 + t_1 + t_2 + \dots + t_p} \frac{\sum_{s=0}^{s_0(\varepsilon)} C_m^s C_{L'-m}^{\ell'-s}}{C_L^\ell},$$

где $L' = L - \sum_{i=1}^{p} t_i - \sum_{i=1}^{p} [t_i \neq D]$, $\ell' = \ell - \sum_{i=1}^{p} [t_i \neq D]$, $s_0(\varepsilon) = \lfloor \frac{\ell}{L}(m + h - \varepsilon k) \rfloor$.

Запишем формулу с помощью гипергеометрического распределения:

$$Q_\varepsilon(A) = \sum_{h=0}^{D} |\omega_h| \sum_{t_1=h}^{D} \sum_{t_2=0}^{D} \dots \sum_{t_p=0}^{D} \frac{1}{1 + t_1 + t_2 + \dots + t_p} \frac{C_{L'}^{\ell'}}{C_L^\ell} H_{L'}^{\ell',m}(s_0(\varepsilon)),$$

Упростим запись, введя дополнительные обозначения $S = \sum_{i=1}^{p} t_i$, $F = \sum_{i=1}^{p} [t_i \neq D]$. Параметр S, при таком определении, будет задавать мощность множества неразличимых алгоритмов.

$$Q_\varepsilon(A) = \sum_{h=0}^{D} |\omega_h| \sum_{t_1=h}^{D} \sum_{t_2=0}^{D} \dots \sum_{t_p=0}^{D} \frac{1}{1+S} \frac{C_{L'}^{\ell'}}{C_L^\ell} H_{L'}^{\ell',m}(s_0(\varepsilon)),$$

где $L' = L - S - F$, $\ell' = \ell - F$, $s_0 = \lfloor \frac{\ell}{L}(m + h - \varepsilon k) \rfloor$.

Теперь можно перейти к суммированию по множеству возможных значений параметров S и F.

$$Q_\varepsilon(A) = \sum_{h=0}^{D} |\omega_h| \sum_{S=h}^{pD} \sum_{F=0}^{p} \frac{R_D^h(S,F,p)}{1+S} \frac{C_{L'}^{\ell'}}{C_L^\ell} H_{L'}^{\ell',m}(s_0(\varepsilon)),$$

Здесь $R_D^h(S,F,p)$ — комбинаторный коэффициент, который зависит от параметров S и F, от числа монотонных цепочек p и от их длины D, а так же от h — минимально-возможного значения параметра S. Численно $R_D^h(S,F,p)$ равен числу способов представить число S в виде суммы p неотрицательных слагаемых $t_1, \dots, t_p$, каждое из которых не превосходит D. При этом ровно F слагаемых должно равняться D, а на первое слагаемое накладывается дополнительное ограничение $t_1 \geq h$.

5.5 Семейство «Полный слой алгоритмов»

Пусть длина полной выборки равна $L+p+q$, длины обучения и контроля — ℓ и k соответственно. Рассмотрим тогда следующее множество алгоритмов. Выделен набор из p объектов $X_p \subset X$, на которых не ошибается ни один алгоритм. Выделен набор из q объектов $X_q \subset X$, на которых ошибаются все алгоритмы. Имеется C_L^m алгоритмов, каждый из которых ошибается равно на $q+m$ объектах. Таким образом перебираются все возможные алгоритмы, m ошибок которых распределено по оставшимся L объектам полной выборки $X_L = X \backslash \{X_p \cup X_q\}$. Определенное выше множество алгоритмов будем называть *полным слоем* и обозначать A_L^m.

Theorem. Группа симметрий полного слоя алгоритмов — это декартово произведение трех симметрических групп, переставляющих всевозможными способами объекты множеств X_p, X_q и X_L:

$$S(A) = S_p \times S_q \times S_L.$$

Theorem. Группа $S(A_L^m)$ действует на полном слое алгоритмов алгоритмов A_L^m транзитивно.

Следовательно, мы опять имеем дело с простой ситуацией, в который слой алгоритмов содержит лишь один класс идентичности алгоритмов. Следовательно, все алгоритмы дают равный вклад в вероятность переобучения $Q_\epsilon(A_L^m) = C_L^m Q_\epsilon(A_L^m, a_0)$. Для вычисления вероятности переобучения одного алгоритма из семейства выделим специфического представителя a_0, который ошибается на объектах $X_m = \{x_1, x_2, \ldots, x_m\} \subset X_L$, и не ошибается на оставшихся L-объектах $X_L \backslash X_m = \{x_{m+1}, \ldots, x_L\}$.

Пусть $v = |X_L \cap X^k|$ (число объектов выборки X^L, попавших в контроль). Рассмотрим две взаимоисключающие ситуации $v < m$ и $v \geq m$.

Во втором случае алгоритм a_0 будет выбран только при условии $X_m \subset X^k$ — все его L-ошибки расположены в контрольной выборки. В первом случае все эти ошибки просто не могут поместиться в контроль. В этой ситуации a_0 будет выбран если v из m его ошибок попало на контроль, а оставшиеся $m-v$ ошибок останутся в обучении.

Рассмотрим оба случая подробнее.

Случай $v < m$**.** В данной ситуации алгоритм a_0 будет выбран, только если макисмально-возможное число его ошибок попало в контроль. Выбрать v ошибок из m можно C_m^v способами. Осталось подсчитать число алгоритмов, ошибающихся на объектах $\{x_1, \ldots, x_v\}$. Их C_{L-v}^{m-v} штук. Как и в прошлом случае параметр v задает множество разбиений выборки X_L на обучение и контроль. Для каждого из этих разбиений из $p+q$ оставшихся объектов необходим выбрать $\ell'' = \ell - (L-v)$ так, что бы наблюдалась ситуация переобучения алгоритма a_0.

Пусть s — это количество объектов из X_q, попавших на обучение. Тогда, согласно гипергеометрическому распределению, переобучение будет наблюдатся в $C_{p+q}^{\ell''} H_{p+q}^{\ell'',q}(s_0) = \sum_{s=0}^{s_0} C_q^s C_p^{\ell''-s}$ случаях, где $s_0 = \frac{\ell}{L}(q - \frac{k}{\ell}m - \epsilon k)$.

Собирая подсчитанные коэффициенты вместе получаем вклад случая $v < m$ в вероятность переобучения:

$$\frac{C_L^m}{C_{L+p+q}^{\ell}} \sum_{v=0}^{m-1} \frac{C_m^v}{C_{L-v}^{m-v}} C_{p+q}^{\ell'} H_{p+q}^{\ell'',q}(s_0).$$

Случай $v \geq m$**.** Во этом случае алгоритм a_0 будет выбран только в том случае, если объекты $X_m = \{x_1, x_2, \ldots, x_m\}$ попали в контроль. Обозначим $t = v - m$ — количество остальных L-объектов, попавших в контроль. Выбрать эти t объектов можно C_{L-m}^t способами. Вычислим мощность множества $E_{A,X}$ для каждого из этих случаев. Для того, что бы алгоритм попал в это множество, необходимо что бы его ошибки не содержались среди выбраных $t+m$ объектов контрольной выборки. Следовательно имеется C_{L-m-t}^m

вариантов расставить его ошибки в допустимые места. Заметим, что все такие алгоритмы принадлежат нашему семейству.

Параметром t задается определенный набор разбиений выборки X_L на обучение и контроль. Остается разбить оставшиеся объекты $X_p \cup X_q$ таким образом, что бы на обучение было выбрано $\ell' = \ell - (L - t - m)$ объектов. Заметим, что это можно делать независимо относительно разбиений выборки X_L. Пусть из ℓ' требуемых объектов s выбрано из множества ошибок X_q. Тогда число разбиений $X_p \cup X_q$, при котором наблюдается переобучение, задается гипергеометрическим распределением $C^{\ell'}_{p+q} H^{\ell',q}_{p+q}(s_1) = \sum_{s=0}^{s_1} C^s_q C^{\ell'-s}_p$, где $s_1 = \frac{\ell}{L+p+q}(q + m - \epsilon k)$.

Таким образом, вклад случая $v \geq m$ в вероятность переобучения имеет вид

$$\frac{C^m_L}{C^\ell_{L+p+q}} \sum_{t=0}^{k-m} \frac{C^t_{L-m}}{C^m_{L-m-t}} C^{\ell'}_{p+q} H^{\ell',q}_{p+q}(s_1).$$

Собирая полученные результаты, приходим к следующей теореме. **Theorem.** Вероятность переобучения полного слоя алгоритмов дается формулой

$$Q_\epsilon(A) = \frac{C^m_L}{C^\ell_{L+p+q}} \sum_{v=0}^{m-1} \frac{C^v_m C^{\ell'}_{p+q}}{C^{m-v}_{L-v}} H^{\ell',q}_{p+q}(s_0) + \frac{C^m_L}{C^\ell_{L+p+q}} \sum_{v=m}^{k} \frac{C^{v-m}_{L-m} C^{\ell'}_{p+q}}{C^m_{L-v}} H^{\ell',q}_{p+q}(s_1), \quad (1)$$

где $s_0 = \frac{\ell}{L}(q - \frac{k}{\ell}m - \epsilon k)$, $s_1 = \frac{\ell}{L+p+q}(q + m - \epsilon k)$, $\ell' = \ell - (L - v)$.

В простом случае $p = q = 0$ можно вывести оценку

$$Q_\epsilon(A) = \begin{cases} \frac{C^m_L C^{k-m}_{L-m}}{C^k_L C^m_k}[m \geq \varepsilon k], \text{ при } m \leq k; \\ \frac{C^m_L C^k_m}{C^k_L C^{m-k}_{L-k}}[m \leq L - \varepsilon\ell], \text{ при } m > k; \end{cases}$$

Легко убедиться, что комбинаторный множитель равен единице в обоих случаях. Следовательно, приходим к простому выражению:

$$Q_\epsilon(A) = \begin{cases} 1, \text{ при } \varepsilon k \leq m \leq L - \varepsilon\ell; \\ 0, \text{ в противном случае.} \end{cases}$$

6 КБШ-оценка вероятности переобучения

Для того, что бы выписать точную оценку вероятности переобучения для конкретного семейства необходимо для каждого алгоритма указать множество условий, налагаемых на разбиение на обучение и контроль, при которых данный алгоритм имеет шанс реализоваться. При этих ограничениях необходимо посчитать число случаев, когда выполнено $[\Delta\nu(a, X) \geq \varepsilon]$. Это число уже может не выражаться в виде простого комбинаторного выражения. В идеале хотелось бы иметь оценку, в которую для каждого алгоритма входит вероятность получить его в результате обучения и вероятность переобучения данного алгоритма на всей выборке, зависящая только от числа ошибок алгоритма.

Подобную оценку можно получить с помощью неравенства Коши-Буняковского-

Шварца.

$$Q_\mu(A) = \frac{1}{C_L^\ell} \sum_{X\in[X]_L^\ell} \sum_{a\in A} \mu(A,X,a)\,[\Delta\nu(a,X) \geq \varepsilon] =$$
$$= \frac{1}{C_L^\ell} \sum_{a\in A} \sum_{X\in[X]_L^\ell} \mu(A,X,a)\,[\Delta\nu(a,X) \geq \varepsilon] \leq$$
$$\leq \sum_{a\in A} \sqrt{\frac{\sum_{X\in[X]_L^\ell} \mu^2(A,X,a)}{C_L^\ell}} \sqrt{\frac{\sum_{X\in[X]_L^\ell} [\Delta\nu(a,X) \geq \varepsilon]^2}{C_L^\ell}} =$$
$$= \sum_{a\in A} \sqrt{\frac{\sum_{X\in[X]_L^\ell} \mu^2(A,X,a)}{C_L^\ell}} \sqrt{H_L^{\ell,m_a}(s)} =$$
$$= \sum_{a\in A} \sqrt{\langle \mu^2(A,X,a)\rangle} \sqrt{H_L^{\ell,m_a}(s)}$$

В итоговую оценку входит среднеквадратичная вероятность получения алгоритма в результате обучения и корень из гипергеометрического распределение на полной выборке. Данный результат легко совместить с теоремой о равном вкладе эквивалентных алгоритмов в вероятность переобучения:

$$Q_\mu(A) \leq \sum_{\omega\in\Omega(A)} |\omega| \sqrt{\langle \mu^2(A,X,a_\omega)\rangle} \sqrt{H_L^{\ell,m_{a_\omega}}(s)}$$

Можно сформулировать этот результат для метода обучения, который возвращает один алгоритм: $\mu : \aleph \times [X]_L^\ell \to \mathbb{A}$:

$$Q_\mu(A) = \frac{1}{C_L^\ell} \sum_{a\in A} \sum_{X\in[X]_L^\ell} [\mu(A,X) = a]\,[\Delta\nu(a,X) \geq \varepsilon] =$$
$$\leq \sum_{a\in A} \sqrt{\frac{\sum_{X\in[X]_L^\ell} [\mu(A,X) = a]^2}{C_L^\ell}} \sqrt{\frac{\sum_{X\in[X]_L^\ell} [\Delta\nu(a,X) \geq \varepsilon]^2}{C_L^\ell}} =$$
$$= \sum_{a\in A} \sqrt{P_a} \sqrt{H_L^{\ell,m_a}(s)} = \sum_{a\in A} \sqrt{P_a H_L^{\ell,m_a}(s)}.$$

7 Граф смежности семейства алгоритмов

Набору алгоритмов можно поставить в соответствие граф. Его вершины соответствуют алгоритмам, а ребра соединяют алгоритмы, в которых вектора ошибок отличаются на единицу. Перестановки объектов сохраняют расстояние между алгоритмами, поэтому граф смежности корректно определен для семейства алгоритмов.

Предостережение. Разным семействам алгоритмов может соответствовать один и тот же граф. Пример такой ситуации возможен уже для унимодальной цепочки длины два:

$$\begin{pmatrix} 0&0&0&0\\ 1&0&0&0\\ 1&1&0&0\\ 0&0&1&0\\ 0&0&1&1 \end{pmatrix} \qquad \begin{pmatrix} 0&0&0&0\\ 1&0&0&0\\ 1&1&0&0\\ 0&0&1&0\\ 0&1&1&0 \end{pmatrix}$$

Именно последнее обстоятельство не позволяет ввести понятие группы симметрии и идентичности алгоритмов на графе. Хотя, разумеется, идентичные алгоритмы в определенном смысле всегда оказываются "одинаково расположеными" на графе семейства алгоритмов.

Задача. Установить связь между группой симметрий семейства алгоритмов и группой автоморфизмов его графа смежности. *Подсказка.* Неверно утверждать, что группа симметрий является подгруппой группы автоморфизмов графа. Дело в том, что группа автоморфизмов определяется как подгруппа группы перестановок *алгоритмов*, а группа симметрий — как подгруппа группы перестановок *объектов*.

Задача?. Привести контрпример к утверждению "вероятность переобучения зависит только от графа, а не от семейства алгоритмов".

Легко построить пример, состоящий из двух несвязных семейств. Например, две монотонных цепочки, расположенных в одном случае на разных объектах, а во втором — на одних и тех же, но в разном направлении. Однако в данном примере если добавить лучший алгоритм — расхождение пропадает.

8 Вычисление групп симметрии

Теорема (о группе симметрий одного алгоритма). Группа симметрий одного алгоритма является прямым произведением групп перестановок, действующих на множестве единиц и множестве нулей данного алгоритма.

$$S(\{a\}) = S_m \times S_{L-m},$$

где m — число ошибок алгоритма.

Теорема (о группе симметрий шара алгоритмов). Группа симметрии шара алгоритмов содержит в качестве подгруппы группу симметрий семейства, состоящего из центра данного шара.

$$S(\{a\}) \subset S(B_r(a))$$

Если шар $B_r(a)$ не содержит хотя бы одну из точек $(0, 0, \ldots, 0)$ и $(1, 1, \ldots, 1)$ указанные выше включение превращается в точное равенство.

Доказательство. Допустим при перестановке $\pi \in S_L$ центр шара остался на месте: $\pi(a) = a$. Рассмотрим произвольный алгоритм $b \in B_r(a)$. Воспользовавшись тем, что перестановка объектов является изометрией на множестве алгоритмов ($\rho(a, b) = \rho(\pi(a), \pi(b))$) получим, что $\pi(b) \in B_r(\pi(a))$. Обратное утверждение получим, если перестановку π заменить на обратную к ней π^{-1}. Таким образом, шар алгоритмов остался на месте.

Теперь пусть семейство не содержит одной из крайних точек гиперкуба (т.е. тождественного нуля или единицы). От противного: допустим существует перестановка $\pi \in S_L$, не оставляющая центр шара a на месте, но переводящая шар в самого себя:

$$\pi(a) \neq a, \text{ но } \pi(B_r(a)) = B_r(\pi(a))$$

Нарисуем друг под другом алгоритмы a и $\pi(a)$, выбрав удобный порядок следования объектов:

кол-во объектов	m	n	k	k
a	0 0 ... 0	1 1 ... 1	0 0 ... 0	1 1 ... 1
$\pi(a)$	0 0 ... 0	1 1 ... 1	1 1 ... 1	0 0 ... 0

Заметим, что при перестановке часть алгоритма осталась неизменной, а в оставшейся части нули заменились на единицы и наоборот. Поскольку количество единиц осталось прежним, число объектов в двух последних колонках равны.

Теперь выпишем несколько неравенств. Пусть для определенности шар $B_r(a)$ не содержит тождественный нуль. Тогда "$n+k \geq r+1$". Выберем первые $\min(m+n, r)$ координат алгоритма a и заменим их на противоположные. Получившийся алгоритм обозначим

через c. Очевидно, что $c \in B_r(a)$. Покажем, что $c \notin B_r(\pi(a))$, и тем самым придем к противоречию.

Для этого достаточно убедиться, что расстояние $\rho(c, \pi(a)) = \min(m + n + 2k, r + 2k)$ превосходит r. А это верно, поскольку каждое выражение под знаком min больше r. ■

Данная теорема позволяет ответить на вопрос о классах идентичности алгоритмов в семействе, состоящего из одного лучшего алгоритма с **нулем** ошибок на полной выборке и его окрестности произвольного размера. Поскольку в данном случае группа симметрий — это вся группа перестановок S_L, получаем что внутри каждого слоя все алгоритмы идентичны друг другу. Однако в случае, когда центр шара не является лучшим алгоритмом в семействе, классы идентичности устроены более сложным образом.

Теорема. Группа симметрии семейства алгоритмов равна пересечению групп симметрии слоев данного семейства.

История изменений:

Период	Изменение
22 апреля — 10 мая, 2009	Теория симметричных семейств алгоритмов
10 — 26 июля, 2009	Явные формулы для связки из монотонных цепочек
22 — 24 сентября, 2009	Явные формулы для полного слоя алгоритмов